An Environment for Distributed Collaboration Among Humans and Software Agents

C. Martin, D. Schreckenghost, P. Bonasso, D. Kortenkamp, T. Milam, and C. Thronesbery
NASA Johnson Space Center / TRACLabs
1012 Hercules
Houston, TX, 77058 USA
1 (281) 461-9525

cmartin@traclabs.com, ghost@ieee.org

ABSTRACT

This paper describes an implemented software prototype for the Distributed Collaboration and Interaction (DCI) system, which helps humans to act as an integrated part of a multi-agent system. Human interaction with agents who act autonomously most of the time, such as a process control agent in a refinery, has received little attention compared to human interaction with agents who provide a direct service to humans, such as information retrieval. This paper describes how *liaison agents* within the DCI system can support human interaction with other agents that are not, by design, human-centric but must be supervised by, or coordinated with, humans. The DCI system provides a step toward future seamless integration of humans and software agents into a cohesive multi-agent system.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multiagent systems*.

General Terms

Design, Human Factors

Keywords

Liaison agents, human-agent interaction, distributed collaboration and interaction

1. INTRODUCTION

Although many of today's software agents are designed to interact with humans, interaction between humans and agents designed primarily for long-term, mostly-autonomous control operations has received relatively little attention. However, humans need to interact with these agents for a variety of reasons including supervisory monitoring, modifying operating goals or constraints, updating information about the environment or systems, maintaining or repairing underlying hardware or software, responding to anomalies, and taking advantage of opportunities. As more software agents are deployed for long-term autonomous operation, the research challenge of enabling agents and humans to work together as a complete system becomes more important. To meet this challenge, research must overcome issues such as limited visibility into, and understanding of, the agent's processing, mismatches between a human's mental models and the agent's implementation models, lack of adequate adjustability for the agent's autonomy, lack of notification to the human about important events at appropriate levels of abstraction, and a basic incompatibility between a human's natural interface modalities and the usual interfaces provided by an autonomous control agent.

We have developed the Distributed Collaboration and Interaction (DCI) system to address these difficulties in human-agent interaction and to create an environment in which humans and mostly-autonomous software agents together can form an integrated multi-agent system. The initial motivation for the DCI system arose from our experiences with deployed intelligent control agents for NASA advanced life support systems. These intelligent control agents monitor and perform process control for regenerative life support systems, which recover usable water or air from the waste products created by biological systems over time. Over months to years of continuous operation of these agents, we discovered many unaddressed needs for human interaction. These lessons learned are detailed in [4]. Through the DCI project, we have addressed and enhanced our understanding of these interaction needs and have begun to formulate solutions. The DCI architecture, as described in this paper, embodies the software capabilities that we have identified as requirements for supporting human interaction with mostly-autonomous software agents over long-term operation.

2. DCI ARCHITECTURE

The DCI approach uses intermediate *liaison agents*, one associated with each human, to mediate between that human and the rest of a multi-agent system. Software agents for the purpose of assisting humans have also been explored by other research [1; 2]. In addition, DCI uses *augmenting software* associated with existing software in the multi-agent system to handle the human-interfacing requirements related to that domain-centric software.

Figure 1 depicts representative elements of a DCI system. The entities with black backgrounds (the human, the life support system and its control agent, and the multi-agent planner) participate in, but are not part of, the DCI environment. The Conversion Assistant for Planning (CAP) and the Event Detection Assistant (EDA) are examples of augmenting software in the DCI system. The CAP is software that augments a planner's ability to interface with human agents (e.g., when an organization uses planning to coordinate anomaly response activities between software and human agents). The EDA monitors data produced by the life support system or its control agent and searches for

AAMAS'03, July 14–18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

patterns in this data that are of interest to the humans for such activities as anomaly analysis and performance assessment. The augmenting software is designed to tightly couple with the domain-specific software and thus off-load much of the human-centric processing. Our approach avoids overburdening resource-limited software agents, whose processing may include time-critical tasks, with tasks that are not directly related to their primary function.

The liaison agents in DCI are called Attentive Remote Interaction and Execution Liaison (ARIEL) agents, in deference to Shakespeare's Tempest character. The beliefs held by an ARIEL agent are managed by its State Management Service (SMS), which takes input from many of ARIEL's other services and creates a coherent state model of the user, including current activity, location, role, schedule, and health. The SMS also interacts with its user by querying for state input (such as schedule acknowledgements), when appropriate. Other services represented in ARIEL's design include:

- *Notification Service (NS):* The NS combines information about the user's current state and roles, organizational policies about information distribution and situation-awareness requirements, and the user's own information preferences to determine if an incoming notice or event is of interest to the user and, if so, how to inform him or her. The prototype NS implementation and the representation of specifications an organization or individual may use to filter or present notices are described in detail in [3].
- *Task Status Service (TSS):* The TSS provides activity tracking and plan management capabilities, such as monitoring for user acknowledgement or evidence that assigned activities have been initiated and providing a source of feedback to the augmented planner about human progress or lack of progress toward achieving a plan.
- *Location Service (LS):* The LS tracks human location, including physical location and cyber location (i.e., whether or not the user is online and which display platforms he or she is currently using).
- *User Interface Service (UIS):* The UIS manages all direct interaction with the user. It invokes different modalities, such as display, pager, or email, to present information in the manner most appropriate to the user's current state and task.
- *Command and Authorization Service (CAS):* The CAS supports the user in remotely issuing directives to physical systems that are otherwise controlled by mostly autonomous agents. The CAS ensures authorized use and command lockouts, and it aids reconfiguration at control transitions.
- *Interruption Handling Service (IHS):* The IHS coordinates the actions of other services (e.g., the Notification Service) to minimize the impact of interruptions on the user's tasks.
- *Interactive Procedure Service (IPS):* The IPS assists the user in temporarily modifying standard operating procedures executed by mostly autonomous software agents.
- *Interactive Event Service (IES):* The IES assists the user in interactively defining new, temporary, operational events for the purpose of adjusting automated monitoring and notification.

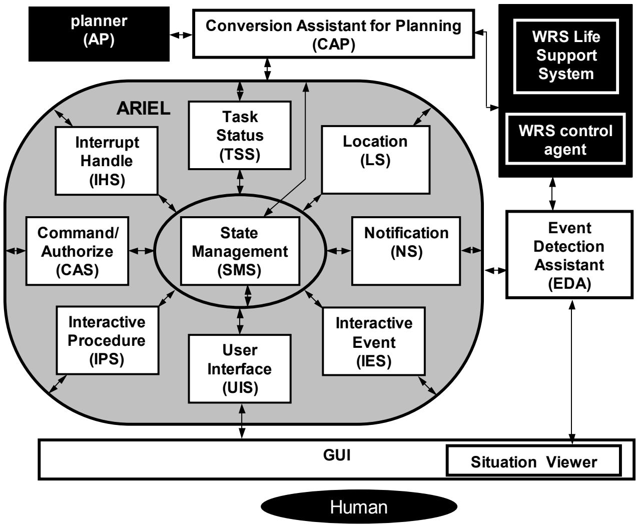

Figure 1. Representative Elements of a DCI System.

Using liaison agents with these services and using augmenting software for existing domain-centric software, DCI supports human interaction with complex, mostly-autonomous software control agents in the context of an integrated multi-agent system. The philosophy behind the DCI system involves creating an environment that supports human collaboration and interaction with software agents and giving humans appropriate tools and interfaces to participate effectively in these interactions. Through the DCI environment, humans can become part of the multi-agent world and act naturally within it.

3. ACKNOWLEDGMENTS

We want to acknowledge the support of Dr Michael Shafto, the manager of the Human-Centered Computing topic in NASA's Intelligent Systems Program, under which this work was done.

4. REFERENCES

[1] Chalupsky, H., Gil, Y., Knoblock, C. A., Lerman, K., Oh, J., Pynadath, D. V., Russ, T. A., and Tambe, M. Electric Elves: Applying Agent Technology to Support Human Organizations. In *Proceedings of Innovative Applications of Artificial Intelligence* (Seattle, WA, 2001). 51-58

[2] Payne, T. R., Sycara, K., and Lewis, M. Varying the User Interaction within Multi-Agent Systems. In *Proceedings of Fourth International Conference on Autonomous Agents* (Barcelona, Catalonia, Spain, 2000). ACM Press. 412-418.

[3] Schreckenghost, D., Martin, C. E., and Thronesbery, C. Specifying Organizational Policies and Individual Preferences for Human-Software Interaction. In *Proceedings of AAAI Fall Symposium on Etiquette for Human-Computer Work (Technical Report FS-02-02)* (North Falmouth, MA, 2002). AAAI Press. 32-39.

[4] Schreckenghost, D., Thronesbery, C., Bonasso, R. P., Kortenkamp, D., and Martin, C. E. Intelligent Control of Life Support for Space Missions. *IEEE Intelligent Systems* 17 (5): 24-31. (2002).